Name ÷ Prof Anand Gharu
Dept ÷ Computer Engg.
Subject ÷ TOC.
Desig. ÷ Assistant Prof.
Date ÷ 03/10/2022.

INFORMATION TECHNOLOGY.

TOC INSEM 2022 SOLUTION

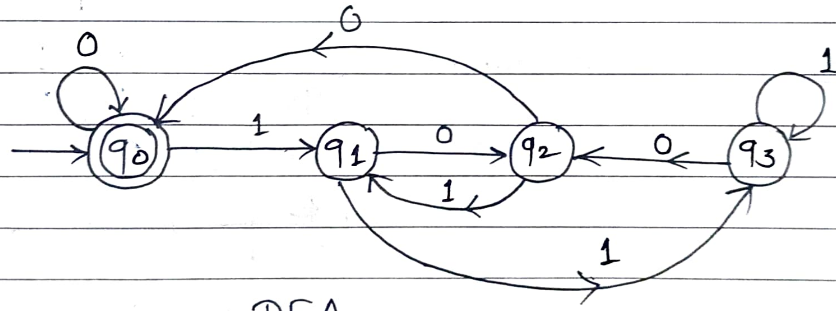
Q. 1 a) Design DFA which accepts Binary nos divisible by 4.

→ 1) Logic ÷ - Input are Binary number $\Sigma = \{0, 1\}$
- it should be divisible of 4

e.g. 00, 100, 1000, 1100, 10000 ... so on.

Remainder		Input.
q_0	0	00, 100, 1000, 1100, 10000 ...
q_1	1	01, 101, 1001, 1101 ...
q_2	2	10, 110, 1010 ...
q_3	3	11, 111, 1011, 1111 ...

2) State Transition Diagram ÷



DFA

3) State Transition Table ÷

	Input	
states	0	1
→ q_0	q_0	q_1
q_1	q_2	q_3
q_2	q_0	q_1
q_3	q_2	q_3

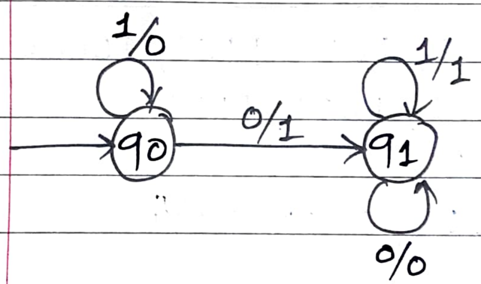
Prepared By. Prof. Anand Gharu

Q 1 b) Design Mealy Machine to increment Binary number by 1. Write down transition table.

→ 1) Logic ÷

- We can read Binary Number from LSB one bit at a time
- We can replace each 1 by 0 until we get 1^{st} 0.
- As we get 1^{st} 0 we will replace it by 1. then keep remaining bit as it is.

2) State Transition Diagram ÷



Mealy Machine

e.g.
0000 → 0001
0001 → 0010
0010 → 0011
0011 → 0100
0100 → 0101
0101 → 0110
0110 → 0111 so on

0 1 1 0
as it is ↓ change
0 1 1 1

3) State Transition table

	Next state			
state	0	o/p	1	o/p
q_0	q_1	1	q_0	0
q_1	q_1	0	q_1	1

Q.1.c] Convert the foll. NFA − ∈ to DFA.

states / Input	δ: ∈	a	b	c
→ p	{p}	{q}	φ	φ
q	{r}	φ	{q}	φ
r*	φ	φ	φ	{r}

→ Step 1 ÷ Find ∈-closure of all states

p = {p, q, r}
q = {q, r}
r = {r}

By. Prof Anand Gharu

Q.1 c) NFA-ε to DFA :-

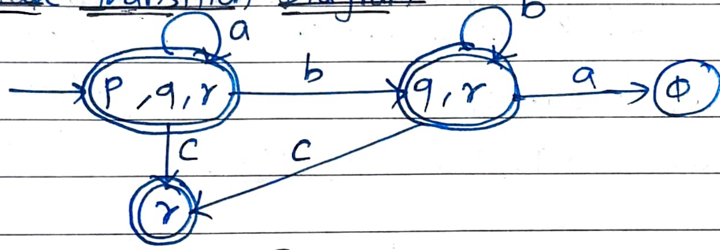
→ Step-2 :-

Take P as start state But In ε-NFA, we have to consider it's ε-closure of P. or any new states.

ε-closure new state	a	b	c	
p, q, r.	p, q, r.	q, r.	r	q, r is new state
q, r	φ	q, r.	r	No new state
r*	φ	φ	r	No new state.

Step 3:

State transition Diagram



Final DFA.

OR

Q.2 a) Define following term with Proper Examples :-

i) Alphabets :-

An alphabet is a finite, non-empty set of symbol. Alphabets is denoted by Σ this symbol.

e.g. Σ = {0, 1} , Σ = {a, b}

ii) String :-

A string is a finite sequence of symbols from alphabet.

e.g. 1010 is string from Binary alphabets.

2345 is string from Decimal alphabets.

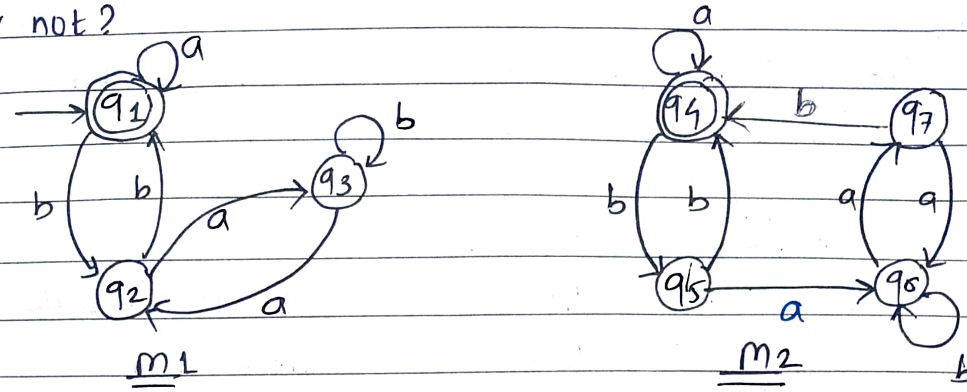
iii) Natural language :-

- Natural language are the language that people speaks likes Marathi, Hindi, English etc.

- These languages are not designed by people.

By Prof. Anand Gharu

Q.2 b) Show whether the foll. automata M1 & M2 are equivalent or not?



→ Step-1 :

Consider start state of M1 & M2 to get new state from M1 & M2. ~~Here~~,

If new state found means equivalent (For all states) [both new states should final or Non-final states)

$\delta(\{q_1, q_4\}, a) = \{q_1, q_4\}$
$\delta(\{q_1, q_4\}, b) = \{q_2, q_5\}$ → new state

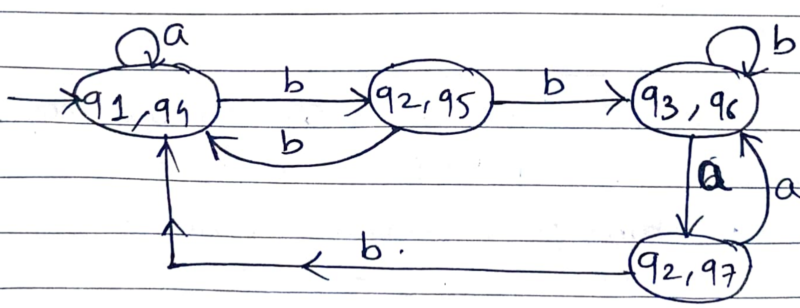
} Check outgoing edges from states.

Step-2 : $\delta(\{q_2, q_5\}, b) = \{q_1, q_4\}$
$\delta(\{q_2, q_5\}, a) = \{q_3, q_6\}$ → new state

Step 3 : $\delta(\{q_3, q_6\}, b) = \{q_3, q_6\}$
$\delta(\{q_3, q_6\}, a) = \{q_2, q_7\}$ → new state

$\delta(\{q_2, q_7\}, a) = \{q_3, q_6\}$
$\delta(\{q_2, q_7\}, b) = \{q_1, q_4\}$

} No new states. so stop.



M3

Every state in M3 has been expanded.
Hence, Construction of DFA is over. 4 states are generated

1) (q_1, q_4) → q_1 & q_4 Both are final states
2) (q_2, q_5) → q_2 & q_5 both are non-final states
3) (q_3, q_6) → Non final states
4) (q_2, q_7) → Non-final states, Thus, M1 & M2 are equivalent.

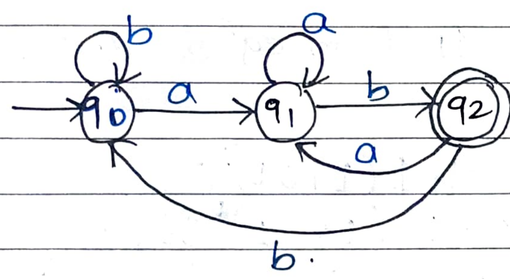
By: Prof. A.N. Ghan

Q.2 c) Construct DFA over the alphabets {a, b} for accepting the strings ending with "ab"

→ 1) Logic :-

- Input are a, b $\Sigma = \{a, b\}$
- string ends with ab.
- e.g. ab, aab, bab, bbab --- so on

2) State Transition Diagram :-

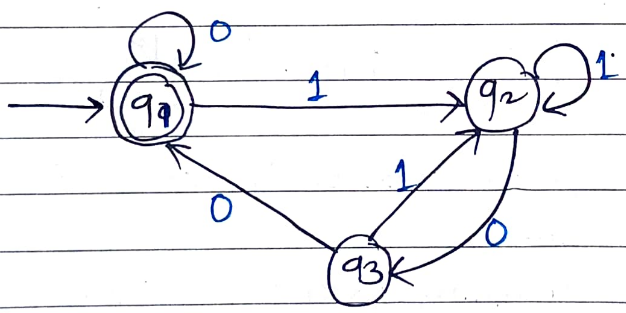


DFA

3) State Transition Table

States	Inputs a	b
→ q_0	q_1	q_0
q_1	q_1	q_2
* q_2	q_1	q_0

Q.3 a) Find the Regular Expression for set of strings recognized by the given FA using Arden's theorem.



→ step1 :- Find equations for all states. & simplify if possible.

$$q_1 = q_1 0 + q_3 0 + \epsilon \quad \text{—— (1)}$$
$$q_2 = q_1 1 + q_2 1 + q_3 1 \quad \text{—— (2)}$$
$$q_3 = q_2 0 \quad \text{—— (3)}$$

Substitute the value of q_3 in eq^n (1) & (2),

We get, $q_1 = q_1 0 + q_2 00 + \epsilon$ —— (4)

$$q_2 = q_1 1 + q_2 1 + q_2 01$$
$$q_2 = q_1 1 + q_2 (1 + 01) \quad \text{—— (5)}$$

From Arden's Theorem

By. Prof-Anand Gharu

Q. 3 a) Arden's Theorem :-

→ From Arden's Theorem

$$q_2 = \underbrace{q_1 1}_{Q} + \underbrace{q_2}_{R}\underbrace{(1+01)}_{P} \quad \text{--- } ⑤$$

$$R = Q \quad R \quad P$$

So, We can Rewrite

$$R = QP^*$$

$$\boxed{q_2 = q_1 1(1+01)^*} \text{ --- } ⑥$$

Substitute value of q_1 from eqⁿ ⑥ in eqⁿ ④,

We get,

$$q_1 = q_1 0 + q_1 1(1+01)^* 00 + \epsilon$$

$$\underbrace{q_1}_{R} = \underbrace{q_1}_{R}\underbrace{[(0+1)(1+01)^* 00]}_{P} + \underbrace{\epsilon}_{Q}$$

By Arden's Theorem.

$$R = RP + Q$$

We can Rewrite it as

$$R = QP^*$$

$$q_1 = \epsilon \cdot [(0+1)(1+01)^* 00]^*$$

$$\boxed{R.E = [(0+1)(1+01)^* 00]^*}$$

b) Determine the Regular Expression over $\Sigma = \{0,1\}$ for the following:

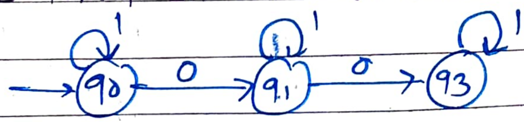
i) All the string Containing exactly two 0's.

ii) All the string do not end with 01

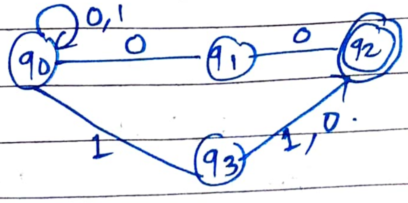
iii) All the string Containing 1 as third character from end.

→

i) $1^* 0 1^* 0 1^*$

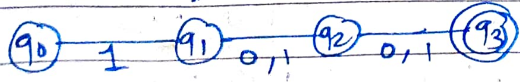


2) $(0+1)^* + (00 + 11 + 10)$



3) $1 \cdot (0+1) \cdot (0+1)$

↓



By Prof Anand Gharu

Q.3 c) Explain the following term

i) Kleene closure

ii) Positive closure

→ 1) Kleene closure :

Given an Alphabet Σ the Kleene closure of Σ is a lang given by

$$\Sigma^* = \underbrace{\Sigma^0}_{\epsilon} \cup \underbrace{\Sigma^1}_{\text{string of length 1}} \cup \underbrace{\Sigma^2}_{\text{string of length 2}} \ldots$$

For Example :

If $\Sigma = \{x\}$, then

$\Sigma^* = \{\epsilon, x, xx, xxx \ldots\}$

If $\Sigma = \{0,1\}$ then

$\Sigma^* = \{\epsilon, 0, 1, 10, 01, 11 \ldots\}$

ii) Positive Closure :

L^+ is a Positive closure of language.

L^+ is the set of all finite string formed by Concatenating word from L.

Any word can be used one / more time.

Example :

If $L = \{a\}$ then

$L^+ = \{a, aa, aaa \ldots\}$

If $L = \{aa, b\}$ then

$L^+ = \{b, aa, bb, bbb, baa \ldots\}$

In this ϵ - (epsilon) input are not allowed)

Prepared By Prof. Anand Gharu

Q.4 a) Explain Any three closure Properties of Regular language.

→ 1) Union of Two Regular languages is regular

2) The Intersection two Regular language is regular

3) The closure operation on a Regular language is regular.

4) The Concatenation of Regular language is regular.

5) The Concatenation of Regular language is Regular

1) Union of Two Regular language is Regular.

→ If L1 & L2 are Regular. then

They have Regular Expression $L1 = L(R_1)$ and
$L2 = L(R_2)$

Then, $L1 \cup L2 = L(R_1 + R_2)$

Thus, We get $L1 \cup L2$ as Regular language.

(Any lang, given by Some Regular Expression is Regular)

2) The closure Operation on Regular language is Regular

→ - If Language L is regular. then it can be expressed as
$L1 = L(R_1^*)$

Thus, for closure operation a lang. can be expressed as a lang. of Regular Expression.

Hence, L is said to be Regular language.

e.g.

$a^* = \{\epsilon, a, aa, aaa \ldots\}$.

3) The Concatenation of Regular Language is Regular.

→ If L1 & L2 are two language. then $L1 \cdot L2$ is regular. In other word Regular language is closed Under Concatenation.

If L1 & L2 are regular then they can be expressed as $L1 = L(R_1)$ and $L2 = L(R_2)$

Then $L1 \cdot L2 = L(R_1 \cdot R_2)$

Thus, we get a Regular language.

Hence, it is proved that Regular lang. is closed Under Concatenation.

By Prof. Anand Gharu.

Q.4 b) What is Regular Expression? Explain in brief the application of Regular Expression?

→ **Regular Expression:**

"The language accepted by Finite Automata can be easily described by Simple expression is called as Regular Expression."

Example:

$$R.E. = a^*$$

$$R.E = (a+b)^* \text{ OR } (0+1)^*$$

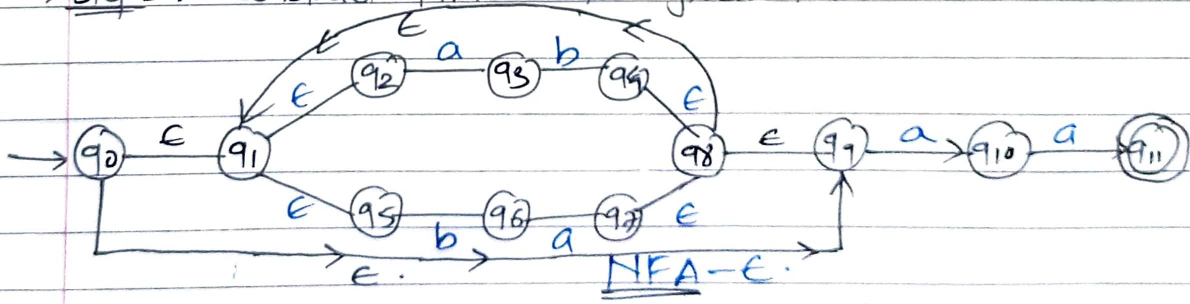
- **Application of Regular Expression:**

1) Regular expression are useful in wide variety of text Processing tasks and more generally word Processing.
2) Common application includes
   - data validation, data wrangling, Simple Parsing etc.
3) It is also useful in Internet Search Engine
4) Verify the structure of string
5) Search / Replace / rearrange part of string.
6) Split a string into tokens

c) Construct a NFA for foll. R.E. using Direct Method:

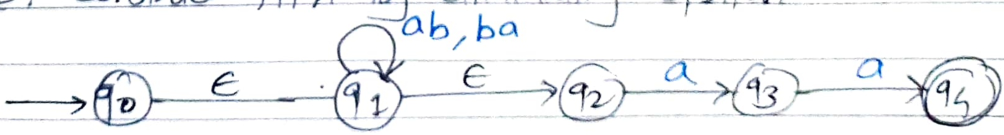
$$R.E. = (ab+ba)^* aa$$

→ **Step 1:** Construct NFA-ε for given R.E.

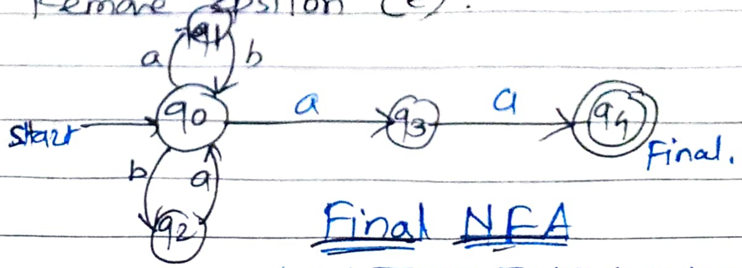


NFA-ε.

Step 2: Construct NFA By eliminating Epsilon.



Step 3: Remove Epsilon (ε).



Final NFA

Prof Anand Ghow

THE END